



**OBI2013**

## **Caderno de Tarefas**

**Modalidade Programação • Nível Júnior, Fase 2**

31 de agosto de 2013

**A PROVA TEM DURAÇÃO DE 3 HORAS**

**Promoção:**



**Sociedade Brasileira de Computação**

**Patrocínio:**



**Fundação Carlos Chagas**

# Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 6 páginas (não contando a folha de rosto), numeradas de 1 a 6. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; e soluções na linguagem Python devem ser arquivos com sufixo *.py*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
  - em Pascal: *readln*, *read*, *writeln*, *write*;
  - em C: *scanf*, *getchar*, *printf*, *putchar*;
  - em C++: as mesmas de C ou os objetos *cout* e *cin*.
  - em Java: qualquer classe ou função padrão, como por exemplo *Scanner*, *BufferedReader*, *BufferedWriter* e *System.out.println*
  - em Python: *read*, *readline*, *readlines*, *print*, *write*
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

# Volume da TV

Nome do arquivo fonte: `volume.c`, `volume.cpp`, `volume.pas`, `volume.java`, ou `volume.py`

Bruno é um menino que gosta muito de ver televisão. No entanto ele se depara com um problema muito chato. Sempre que começa um novo programa no canal preferido dele, a TV Nlogônia, acontece de o volume do som deste programa estar diferente do anterior, às vezes com volume menor, outras vezes com volume maior. Quando está com volume menor, ele aumenta o volume pressionando uma quantidade de vezes seguidas o botão de aumentar para o volume ficar ideal; a mesma coisa acontece quando está um volume maior, e ele diminui o volume pressionando alguma quantidade de vezes seguidas o botão de diminuir o volume para ficar com o volume que ele goste no momento.

O aparelho de TV dele tem umas peculiaridades: ele possui volume mínimo, com valor 0 (também chamado de mudo), e volume máximo, com valor 100. A TV nunca ultrapassa os volumes máximo e mínimo. Por exemplo, se o volume já estiver no máximo e ele pressionar o botão de aumentar o som, o volume não se altera. Da mesma forma, se o volume estiver no valor mínimo e ele pressionar o botão de diminuir o som, o volume não se altera.

Agora Bruno quer sua ajuda: ele lembra qual era o volume inicial da TV, e quantas vezes ele pressionou cada botão. Mas, como foram várias mudanças de volume, ele não sabe qual é o volume atual da TV. Por isso, pediu que você o ajude a calcular qual é o volume atual, dados o volume inicial e a lista de trocas de volume que ele realizou.

## Entrada

A primeira linha da entrada contém dois números inteiros  $V$  e  $T$ , que indicam, respectivamente, o volume inicial e o número de trocas de volume.

A segunda linha contém  $T$  números inteiros  $A_i$  que mostram as modificações de volume realizadas, na ordem em que estas modificações foram feitas. O primeiro número indica a primeira modificação de volume, o segundo número indica a segunda modificação, e assim por diante. Para cada modificação, um número maior do que zero significa quantas vezes Bruno pressionou o botão de aumentar o som; um número menor do que zero significa quantas vezes ele pressionou o botão de diminuir o som. Ou seja, se o número é igual a 5, significa que nessa modificação ele pressionou cinco vezes o botão de aumentar o som; se o número é igual a  $-3$ , significa que nessa modificação ele pressionou o botão de diminuir o som três vezes.

## Saída

Seu programa deve imprimir apenas uma linha, contendo apenas um inteiro  $F$ , que indica qual o volume atual da TV após as mudanças de volume.

## Restrições

- $0 \leq V \leq 100$
- $0 \leq T \leq 1000$
- Para cada variação de volume  $A_i$ ,  $-100 \leq A_i \leq 100$

**Exemplos**

<b>Entrada</b>	<b>Saída</b>
50 4 11 20 -15 -13	53

<b>Entrada</b>	<b>Saída</b>
50 5 30 30 30 40 -78	22

# Soma de Frações

Nome do arquivo fonte: `fracoes.c`, `fracoes.cpp`, `fracoes.pas`, `fracoes.java`, ou `fracoes.py`

Joãozinho está aprendendo a somar frações na escola e quer sua ajuda para escrever um programa que dadas duas frações imprima a soma delas em sua forma irredutível. Assim ele vai poder conferir as respostas dos exercícios que está fazendo.

A forma irredutível de uma fração é quando o divisor (número de baixo) é o menor possível. Por exemplo,  $\frac{10}{3}$  é uma fração irredutível, pois 10 e 3 não têm nenhum divisor em comum. Mas  $\frac{10}{6}$  não é, pois ela pode ser simplificada para  $\frac{5}{3}$ , dividindo-se 10 e 6 por 2.

Dados quatro inteiros  $a$ ,  $b$ ,  $c$ ,  $d$ , escreva um programa que calcule  $\frac{a}{b} + \frac{c}{d}$  na sua forma irredutível.

## Entrada

A única linha da entrada contém quatro inteiros  $a$ ,  $b$ ,  $c$ ,  $d$ , respectivamente dividendo e divisor da primeira fração e dividendo e divisor da segunda fração.

## Saída

Seu programa deve imprimir uma única linha contendo dois inteiros, dividendo e divisor da fração irredutível formada pela soma das duas frações dadas.

## Restrições

- $1 \leq a, b, c, d \leq 100$

## Exemplos

<b>Entrada</b> 2 3 7 3	<b>Saída</b> 3 1
<b>Entrada</b> 7 5 3 2	<b>Saída</b> 29 10

# Plantação

*Nome do arquivo fonte:* `plantacao.c`, `plantacao.cpp`, `plantacao.pas`, `plantacao.java`, *ou* `plantacao.py`

A N-logônia é uma região com um clima muito intenso e variável, onde em questão de poucos dias é possível observar uma forte seca, seguida de uma intensa estação de chuvas. O Seu João tem uma plantação de obilina, uma fruta típica e muito apreciada na região, o que a torna muito valiosa. A obilina, entretanto, é muito suscetível a mudanças climáticas, de forma que é difícil prever quanto desta fruta será colhido durante a safra.

Observou-se que as árvores de obilina seguem as seguintes regras:

- As árvores produzem frutas todos os dias, exceto quando elas morrem;
- As árvores mortas não produzem frutas, e infelizmente, mesmo que volte a chover, continuam mortas;
- Se choveu na noite anterior, a árvore produz uma fruta a mais que no dia anterior;
- Se estiou na noite anterior, a árvore produz uma fruta a menos que no dia anterior; e
- Uma árvore morre se não produzir nenhuma fruta.

O Seu João deseja vender toda a obilina produzida para uma grande rede de mercados local, mas para isso, precisa saber exatamente quantas frutas de obilina ele colherá durante a safra.

Para ajudar o Seu João nesta tarefa, você deve escrever um programa que, dada a previsão do tempo para cada noite do período da safra, e quantas frutas cada árvore do Seu João produziu no dia anterior ao início da safra, determine quantas obilinas serão colhidas durante a safra.

Por exemplo, considerando apenas um pé de obilina, se a safra dura dois dias, choveu durante duas noites, e o pé de obilina produziu 3 frutos antes de começar a safra, a produção total da safra será de 9 frutas: 4 no primeiro dia da safra, e 5 no segundo dia.

## Entrada

A primeira linha da entrada contém dois inteiros,  $N$  e  $K$ , respectivamente o número de dias que dura a safra, e o número de árvores que o Seu João possui.

A segunda linha contém  $K$  inteiros  $a_i$  indicando quantas frutas foram produzidas no dia anterior ao início da safra por cada uma das  $K$  árvores.

A linha seguinte contém  $N$  letras separadas por um espaço em branco. Cada uma das letras indica se choveu ou se estiou durante a noite respectiva: a primeira letra se refere à primeira noite, a segunda letra se refere à segunda noite, e assim por diante. Se a letra for um 'C', indica que choveu aquela noite chuvosa, e se for um 'E', indica que estiou (ou seja, não choveu).

## Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando o número de frutas que serão produzidas pela plantação do Seu João.

## Restrições

- $1 \leq N \leq 100\,000$
- $1 \leq K \leq 100\,000$
- $1 \leq a_i \leq 100$  para todo  $i$

**Informações sobre a pontuação**

- Em um conjunto de casos de teste valendo 70 pontos,  $N \leq 1000$  e  $K \leq 1000$ .
- Em um conjunto de casos de teste valendo 70 pontos, a resposta não excederá 1.000.000.000.

**Exemplos**

<b>Entrada</b>	<b>Saída</b>
3 2 1 2 C E C	13

<b>Entrada</b>	<b>Saída</b>
5 3 2 3 1 E E E C C	4